

## 一种网络时延矩阵分布式自适应重建算法

王聪 张凤荔\* 王瑞锦 李敏 杨晓翔  
(电子科技大学计算机科学与工程学院 成都 611731)

**摘要:** 该文基于互联网时延矩阵的近似稀疏性, 通过给定重建矩阵的零范数先验估计, 讨论了不完整时延矩阵在完全去中心化环境下的填充问题。首先, 将该问题转化为一对耦合凸优化问题, 并进行轮转求解; 然后, 针对次梯度下降求解算法中存在的计算代价过高与泛化能力不足的问题, 提出了搜索上界倍增的自适应分布式矩阵重建 (ADMC) 算法, 并引入不同的损失函数作为重建误差评价准则, 以提升算法的适应能力。实验证明, 在不增加测量与通信负载的前提下, ADMC 能够在不损失精度的情况下显著降低计算代价, 同时, 多种损失函数的引入也提升了算法的鲁棒性。

**关键词:** 计算机网络; 时延估计; 矩阵重建; 稀疏模型; 网络测量; 最优化

**中图分类号:** TP393.1

**文献标识码:** A

**文章编号:** 1009-5896(2014)04-0840-07

**DOI:** 10.3724/SP.J.1146.2013.00960

## An Adaptive Distributed Completion Algorithm for Network Latency Matrix

Wang Cong Zhang Feng-li Wang Rui-jin Li Min Yang Xiao-xiang

(School of Computer Science & Engineering, University of Electronic Science and Technology of China,  
Chengdu 611731, China)

**Abstract:** On the basis of the low-rank characteristic of the Internet latency matrix, the in-complete latency matrix completion problem in full-decentralized environment is studied through setting a priori estimation of the  $l_0$  norm of this matrix. First, the problem is componentized into a couple of convex optimization problems, thus it can be solved by alternative direction method. Then, to achieve low computation cost along with well generalization, an Adaptive Distributed Matrix Completion (ADMC) algorithm is proposed. ADMC doubles the upper-bound of the iterative step size searching area, and introduces several kinds of loss functions as the latency estimation error measures. Experiments show that, without losing any accuracy, ADMC reduces the computation cost significantly without any additional measurement or communication cost, and the introduced various loss functions also improve the robustness of the algorithm.

**Key words:** Computer network; Latency estimation; Matrix completion; Sparse model; Network measurement; Optimization

### 1 引言

互联网环境下的许多重要应用, 包括内容分发网络(CDN)<sup>[1]</sup>, 电子竞技<sup>[2,3]</sup>, 云计算<sup>[4,5]</sup>等, 其性能都强烈依赖于网络时延矩阵的感知与获取。点对点的测量虽然能够精确填充时延矩阵的任一元素, 但其测量复杂度达到  $O(n^2)$ , 难以扩展到大规模网络应用。如何利用有限的测量数据尽可能精确地恢复时延矩阵, 已引起了广泛关注。

早期关于非测量方式的时延矩阵重建研究将节点嵌入到特定的度量空间, 以节点在度量空间内的距离拟合节点间的真实时延, 此类研究成果通常被归为网络坐标系统(Network Coordinate System, NCS)<sup>[6]</sup>。虽然 NCS 能够满足一定程度上的性能需求, 也已在工业环境下得到部分应用, 然而互联网中广泛存在的非对称路由和非最短路径路由现象却违背了度量空间的定义, 且 NCS 涉及的非凸优化问题也使其难以避免局部极小化现象。近期研究证明时延矩阵具备明显的近似稀疏性, 即矩阵仅有少数的特征值具有较大的模<sup>[7]</sup>。文献[8]已就这种稀疏特征及其与 NCS 度量空间的维度关系进行了理论上的深入分析与阐述, 从而将 NCS 归结为某种意义上

2013-07-04 收到, 2013-11-15 改回

国家自然科学基金(61133016), 广东省产学研重点项目(2012B091000054)和中央高校基本科研业务费(ZYGX2010J077)资助课题

\*通信作者: 张凤荔 fzhang@uestc.edu.cn

的时延矩阵稀疏逼近模型。根据 Candes 等人<sup>[9]</sup>的研究, 尽管时延矩阵重建是一个欠定问题, 但在附加必要的稀疏性约束之后, 该问题可以高概率得到唯一的可行解。这种约束一般为矩阵的零范数或迹范数的极小化约束, 以限制重建模型的复杂程度。然而在分布式环境下, 如对等网络或分布式控制系统, 抽取一个全局一致的时延矩阵映像通常是困难的。一个折中的方法是定义一个可能满足稀疏性约束的时延矩阵零范数先验估计, 并将时延矩阵的左右特征向量分布在全网中共同维护, 使得每个参与计算的节点均持有一个行向量和列向量, 通过向量的点积运算, 任一节点都可以补全与其相关的元素。与基于内积运算的 NCS 模型相比, 这种方法具备显著良好的特性: 一是容易转化为凸优化问题求解; 二是容易处理非最短路径路由引起的三角违例 (Triangle Inequality Violations, TIVs) 现象; 三是不要重建矩阵严格对称, 因而容易处理非对称路由问题。基于这种思想, 文献[10]首先提出了一种原型算法 IDES, 但时延空间可能存在的多子流形特征<sup>[11]</sup>, 以及先验估计的不精确性, 使得向量维护和更新过程中难以避免病态矩阵的求逆运算, 导致计算精度难以满足要求<sup>[12]</sup>, 因而一直未能引起足够的重视。直到 Liao 等人<sup>[13]</sup>通过引入正则化算子实现了基于最小均方误差有偏估计的 DMF 算法, 重建算法的鲁棒性才得到实质性的提升。而 Phoenix 算法继承了正则化处理的思路, 并通过非负约束解决了重建矩阵中的负值元素问题<sup>[14]</sup>。最近, 包括  $l_1$  损失函数和 Huber 损失函数在内的兼顾鲁棒性和异常数据处理能力的误差优化准则也开始引起关注<sup>[15,16]</sup>。但受限于目标函数的非光滑性和向量的非负约束, 求解此类损失函数通常仅依赖于梯度下降法, 收敛速度并不令人满意。

本文提出一种支持多种误差评价准则的时延矩阵重建 ADMC 算法。通过引入伸缩因子实现对不同误差评价准则和生命周期不同阶段下梯度下降步长的自适应搜索, 在不损失计算精度的前提下, 以较低的计算代价提升模型的泛化能力。

## 2 时延矩阵分布式重建的自适应迭代算法

时延矩阵分布式重建的基本思想是, 一个包含  $P$  个节点的网络存在  $P \times P$  的时延矩阵  $\mathbf{D}$ , 其中的任一元素  $d_{i,j}$  代表节点  $i$  到节点  $j$  的传输时延, 受到网络规模和计算资源的限制, 一般情况下  $\mathbf{D}$  会因为部分数据的缺失而转变为不完整矩阵  $\mathbf{D}'$ , 时延矩阵重建的目的在于利用  $\mathbf{D}'$  提供的有限信息, 生成一个完整矩阵  $\hat{\mathbf{D}}$ , 使之成为  $\mathbf{D}$  的良好拟合, 全局一致的

不完整矩阵稀疏逼近已得到深入的研究<sup>[17,18]</sup>。考虑到时延矩阵的近似稀疏性, 这种思路显然是合理的。但在完全去中心化的环境下, 时延矩阵一致映像的抽取通常是难以完成的。一般而言, 节点容易协商得到一致的重建矩阵  $\hat{\mathbf{D}}$  的零范数先验估计  $N$ , 以确保稀疏性。此时零范数的极小化约束可松弛为等式约束, 即求解如式(1)的约束优化问题:

$$\min_{\hat{\mathbf{D}}} \left( L \left( P_{\Omega} \left( \hat{\mathbf{D}} - \mathbf{D}' \right) \right) \right), \quad \text{s.t.} \quad \|\hat{\mathbf{D}}\|_0 = N \quad (1)$$

其中  $\Omega$  是  $\mathbf{D}'$  所有未缺失元素的下标子集,  $P_{\Omega}$  是  $\Omega$  上的投影算子, 即只取  $\hat{\mathbf{D}} - \mathbf{D}'$  在  $\Omega$  中的对应元素进行计算;  $L(\cdot)$  是预定义的误差损失函数;  $\|\cdot\|_0$  计算矩阵的零范数, 即矩阵的非零特征值个数。易知  $\hat{\mathbf{D}}$  可表示为两个  $P \times N$  的系数矩阵  $\mathbf{U}$  和  $\mathbf{V}$  的乘积:

$$\hat{\mathbf{D}} = \mathbf{U}\mathbf{V}^T \quad (2)$$

于是  $d_{i,j}$  可用  $\hat{\mathbf{D}}$  的对应元素  $\hat{d}_{i,j}$  拟合:

$$d_{i,j} \approx \hat{d}_{i,j} = \mathbf{u}_i \cdot \mathbf{v}_j^T \quad (3)$$

由于  $\mathbf{U}$  或  $\mathbf{V}$  行向量之间的不存在交集元素, 这使其易分布于全网共同进行维护: 令任一节点  $i$  持有  $\mathbf{U}$  和  $\mathbf{V}$  中对应的行向量  $\mathbf{u}_i$  和  $\mathbf{v}_i$ , 则节点只需获得目标节点持有的向量, 便可计算与对方的近似时延, 从而大大减少了网络测量带来的资源开销。对应地, 式(1)可分解为如式(4)两个耦合子问题的联立:

$$\left. \begin{aligned} \min_{\mathbf{u}_i} \left\{ \sum_{(i,j) \in \Omega} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \right\} \\ \min_{\mathbf{v}_i} \left\{ \sum_{(j,i) \in \Omega} l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \right\} \end{aligned} \right\} \quad (4)$$

式(4)也可加入非负约束以避免  $\hat{\mathbf{D}}$  出现负值元素。虽然式(1)的原始优化问题的凸性无法得到严格确保, 但当  $l(\cdot)$  为凸函数时, 式(4)两个子问题却是凸优化问题, 具备良好的求解性质。考虑到数值解的鲁棒性问题, 可在式(4)中引入正则化因子, 得到式(5):

$$\left. \begin{aligned} \min_{\mathbf{u}_i} \left\{ \lambda \mathbf{u}_i \mathbf{u}_i^T + \sum_{(i,j) \in \Omega} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \right\} \\ \min_{\mathbf{v}_i} \left\{ \lambda \mathbf{u}_j \mathbf{u}_j^T + \sum_{(j,i) \in \Omega} l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \right\} \end{aligned} \right\} \quad (5)$$

正则化项  $\lambda \mathbf{u}_i \mathbf{u}_i^T$  和  $\lambda \mathbf{u}_j \mathbf{u}_j^T$  的引入并不改变问题的凸性。即使在去中心化环境下该问题也很容易通过基于方向轮转的次梯度下降法 (Alternative Dimension Sub-Gradient Method) 交替更新  $\mathbf{u}_i$  和  $\mathbf{v}_i$ :

$$\left. \begin{aligned} \mathbf{u}_i^{(t+1)} &= (1 - \eta_u \lambda) \mathbf{u}_i^{(t)} + \eta_u \sum_{(j,i) \in \Omega} \frac{\partial l(\mathbf{u}_i)}{\partial \mathbf{u}_i} \\ \mathbf{v}_i^{(t+1)} &= (1 - \eta_v \lambda) \mathbf{v}_i^{(t)} + \eta_v \sum_{(j,i) \in \Omega} \frac{\partial l(\mathbf{v}_i)}{\partial \mathbf{v}_i} \end{aligned} \right\} \quad (6)$$

其中  $\eta_u$  和  $\eta_v$  是梯度下降步长。当引入不同的损失函数作为误差评价准则时, 式(6)有不同的表现形式。

当取  $l_1$  损失函数作为误差评价准则时, 有

$$\left. \begin{aligned} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) &= \text{abs}(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \\ l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) &= \text{abs}(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \end{aligned} \right\} \quad (7)$$

此时, 有

$$\left. \begin{aligned} \frac{\partial l(\mathbf{u}_i)}{\partial \mathbf{u}_i} &= -\text{sign}(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \mathbf{u}_i \\ \frac{\partial l(\mathbf{v}_i)}{\partial \mathbf{v}_i} &= -\text{sign}(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \mathbf{v}_i \end{aligned} \right\} \quad (8)$$

当取  $l_2$  损失函数作为误差评价准则时, 有

$$\left. \begin{aligned} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) &= (d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T)^2 \\ l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) &= (d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T)^2 \end{aligned} \right\} \quad (9)$$

此时, 有

$$\left. \begin{aligned} \frac{\partial l(\mathbf{u}_i)}{\partial \mathbf{u}_i} &= -2(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \mathbf{u}_i \\ \frac{\partial l(\mathbf{v}_i)}{\partial \mathbf{v}_i} &= -2(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \mathbf{v}_i \end{aligned} \right\} \quad (10)$$

当取 Huber 损失函数作为误差评价准则时, 有

$$\left. \begin{aligned} l(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) &= \begin{cases} \varepsilon \times \text{abs}(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) - \frac{1}{2} \varepsilon^2, \\ (d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \geq \varepsilon \\ \frac{1}{2} (d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T)^2, \text{ 其它} \end{cases} \\ l(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) &= \begin{cases} \varepsilon \times \text{abs}(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) - \frac{1}{2} \varepsilon^2, \\ (d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \geq \varepsilon \\ \frac{1}{2} (d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T)^2, \text{ 其它} \end{cases} \end{aligned} \right\} \quad (11)$$

此时, 有

$$\left. \begin{aligned} \frac{\partial l(\mathbf{u}_i)}{\partial \mathbf{u}_i} &= \begin{cases} \varepsilon \times \text{sign}(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \mathbf{u}_i, (d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \geq \varepsilon \\ -(d_{i,j} - \mathbf{u}_i \cdot \mathbf{v}_j^T) \mathbf{u}_i, \text{ 其它} \end{cases} \\ \frac{\partial l(\mathbf{v}_i)}{\partial \mathbf{v}_i} &= \begin{cases} \varepsilon \times \text{sign}(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \mathbf{v}_i, (d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \geq \varepsilon \\ -(d_{j,i} - \mathbf{u}_j \cdot \mathbf{v}_i^T) \mathbf{v}_i, \text{ 其它} \end{cases} \end{aligned} \right\} \quad (12)$$

根据文献[19]的定理 2.1.6, 当损失函数为凸函数时, 可保证式(5)为凸优化问题, 此时式(5)的优化过程与此类同, 限于篇幅在此不予讨论。接下来讨论式(6)的求解。显然地, 系统生命周期的不同阶段对  $\eta_u$  和  $\eta_v$  的要求是不同的: 在系统生成初期或网络拓扑结构发生突变时, 需要以较大的步长快速收敛到最优值附近; 而当系统进入到稳定状态时, 需要

以较小的步长进行逐步求精, 并能感知和快速适应随时可能发生的拓扑突变, 如蜂集态(flash-crowd)等。对生命周期不同阶段需求进行折中, DMFSGD 算法<sup>[16]</sup>提出了一种基于折半查找的步长搜索策略, 针对不同的损失函数的搜索上界  $\eta_{\max}$ , 通过实验给出了不同的定义。但该策略的缺点也显而易见: 当  $\eta_{\max}$  较大时, 算法在稳定状态下须经过多轮查找才能得到恰当的值; 而当  $\eta_{\max}$  较小时, 算法难以在可接受的时间内收敛到稳定状态。我们注意到, 系统生命周期的不同阶段存在一定的延续性, 即后一时刻的系统状态与前一时刻往往不会有明显的差异, 换言之, 相邻两个时刻的迭代步长通常并不是条件独立的。基于这种假设, ADMC 算法利用步长搜索的历史信息提出了一种基于上界倍增的步长搜索策略, 以上一轮迭代成功检索的步长因子的某个倍数为搜索上界, 进行折半查找, 从而达到降低计算代价的目的。整个 ADMC 算法的实现伪代码如表 1 所示。

ADMC 算法中, 搜索上界的倍增因子定义为 2。当系统刚刚建立, 或发生拓扑突变时, 该值足以确保迭代步长呈指数级增长, 快速适应新的网络拓扑。表 1 展现的细节显示, 与 DMFSGD 算法相同, ADMC 算法的计算复杂度同样为  $O(m \times n)$ , 其中  $n$  是节点在构造优化函数时选取的参考节点数;  $m$  是

表 1 ADMC 重建算法的伪代码

```

while get ( $d_{i,j}, d_{j,i}, \mathbf{u}_j, \mathbf{v}_j$ )
  Update Neighbor sets;
   $l_u = l(\mathbf{u}_i); l_v = l(\mathbf{v}_i); \eta_u = 2\eta_u; \eta_v = 2\eta_v; t = 0;$ 
  while true
    Update  $\mathbf{u}_i$  by equation (6);
    if  $l(\mathbf{u}_i^{(t)}) < l_u$ 
       $\mathbf{u}_i = \mathbf{u}_i^{(t)};$ 
      break;
    else
       $\eta_u = \eta_u/2; t = t + 1;$ 
    end if
  end while
   $t = 0;$ 
  while true
    Update  $\mathbf{v}_i$  by equation (6);
    if  $l(\mathbf{v}_i^{(t)}) < l_v$ 
       $\mathbf{v}_i = \mathbf{v}_i^{(t)};$ 
      break;
    else
       $\eta_v = \eta_v/2; t = t + 1;$ 
    end if
  end while
  return  $\mathbf{u}_i \mathbf{v}_i;$ 
end while

```

梯度下降步长搜索的轮数。可见 ADCM 算法的测量与计算代价与网络规模无关，同样适用于海量节点在线的网络应用。而相对于 DMFSGD 算法，ADMC 基于历史信息的步长搜索策略通常可以得到较小的  $m$ ，因而其计算代价也显著较低。

### 3 仿真实验

为了检验模型的速度、精度与计算代价，本文以 DMFSGD 算法作为基准算法与 ADCM 算法进行性能对比。所有的算法均假定邻居节点数为 60 个。实验以 Planetlab 数据集<sup>[20]</sup>为仿真基础，该数据集测量了 226 个 Planetlab 节点之间 4h 内的传输时延。在所有实验中，均取重建矩阵的零范数先验估计  $N=5$ ，正则化因子  $\lambda=1$ 。

首先定义距离计算相对误差(Relative Error, RE)以评价系统性能，定义如式(13)：

$$RE_{i,j} = \text{abs}(d_{i,j} - \hat{d}_{i,j}) / \hat{d}_{i,j} \times 100\% \quad (13)$$

与之相关地，引入相对误差的 90%分位数(Ninetieth Percentile Relative Error, NPRE)作为算法整体性能评价指标<sup>[21]</sup>。

#### 3.1 $l_1$ 损失函数性能分析

当基于  $l_1$  损失函数进行矩阵重建时，取推荐值  $\eta_{\max} = 0.01$ ，可得系统性能如图 1 所示。其中，图 1(a)是每一轮迭代中步长折半查找的次数，决定了算法的计算代价。可以看出，在系统的初始阶段，DMFSGD 算法基本无须折半查找，将搜索上界代入计算即可生成损失函数非递增迭代序列。这表明，由于折中得到的搜索上界并非此时的最佳步长。这一现象在图 1(b)和图 1(c)中也得到体现。如图 1(b)所示，以 ADCM 算法搜索得到的步长为对比，当迭

代步长在 0.15 左右时，梯度下降算法才能得到相对令人满意的收敛速度。图 1(c)是系统初始阶段的 NPRE 演进曲线。得益于步长的自适应搜索策略，ADMC 的收敛速度显然较 DMFSGD 更快。需要指出的是，由于向量的初始化策略和  $l_1$  损失函数必然导致的误差截断，使得在系统的起始阶段无论 ADCM 抑或 DMFSGD 都存在一小段“冰冻”时期。但显然 ADCM 的反应更为敏捷，也有较高的收敛速度。由图 1(a)可以看到：随着时延矩阵重建算法逐渐收敛到稳定的状态，DMFSGD 算法的计算代价随之升高。每轮迭代中，DMFSGD 平均需要搜索 6~7 次才能查找到恰当的迭代步长；反观本文提出的 ADCM 算法，每轮迭代中折半查找次数始终维持在 1.95~2.05 次之间，计算代价较 DMFSGD 降低约 65%~70%。这一趋势在图 1(b)中得到印证：当系统趋于稳定时，恰当的迭代步长稳定在  $4 \times 10^{-4}$  左右，恰好是 DMFSGD 算法进行 6~7 次折半查找得到的值；而完成同样的工作 ADCM 通常仅需两次折半查找即可。虽然 ADCM 算法的计算代价大幅降低，但计算性能并未随之产生下降。图 1(c)中明显看出，当系统收敛时，ADMC 和 DMFSGD 算法得到极为相似的 NPRE 演进曲线。深入观察此时的矩阵填充相对误差累积分布如图 1(d)，可见两种算法的累积分布曲线几乎完全重合。由于凸优化问题仅存在一个极优解暨全局最优解，两种不同的算法收敛到同一个解是必然的。

#### 3.2 $l_2$ 损失函数性能分析

当以  $l_2$  损失函数为评价准则时，取推荐值  $\eta_{\max} = 0.001$ ，可得系统性能如图 2。不同于  $l_1$  损失函数， $l_2$  损失函数的迭代步长选取须考虑重建误差的模，

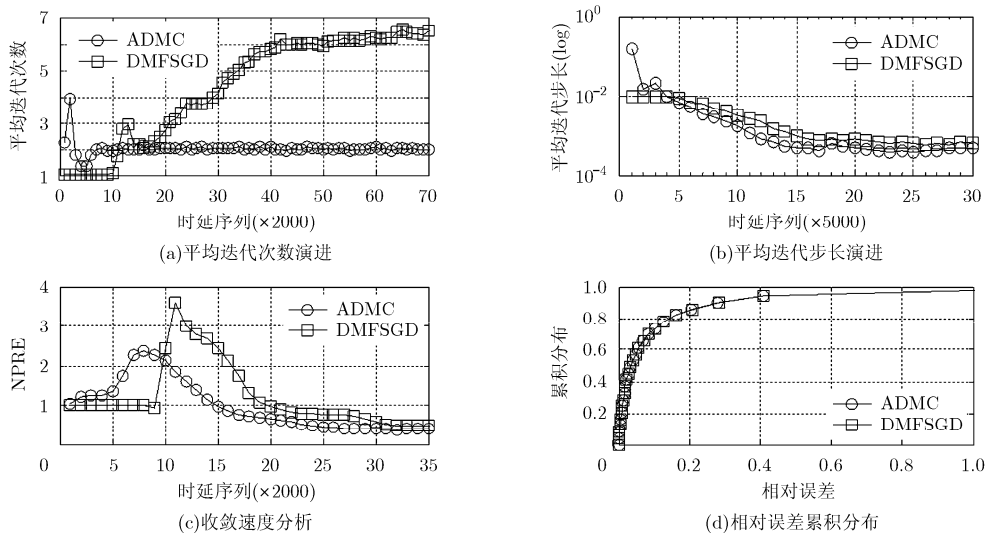
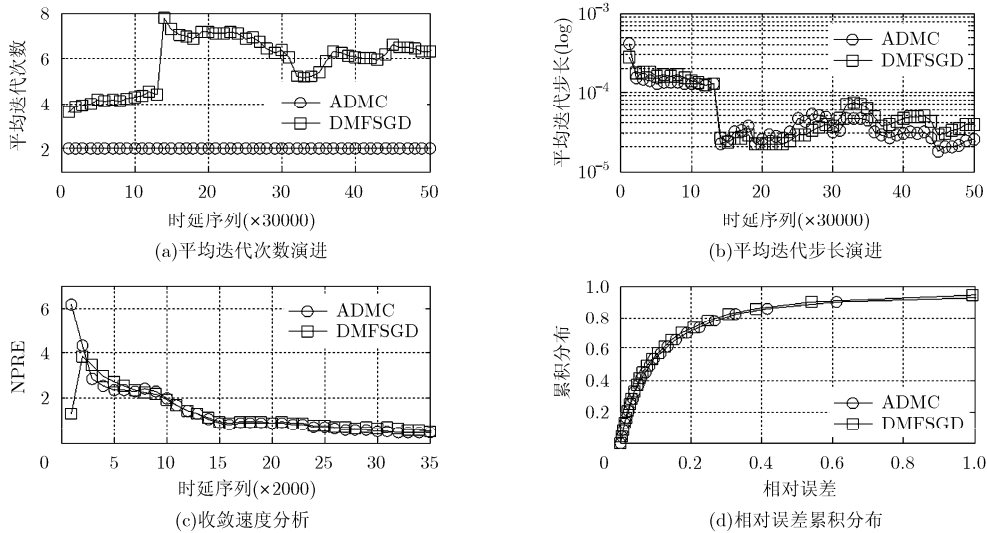


图 1  $l_1$  损失函数算法性能

图 2  $l_2$  损失函数算法性能

因此搜索上界较  $l_1$  损失函数更低, 受网络抖动的影响却大得多。动态环境下的步长平均搜索次数如图 2(a)所示。可以看出, 受数据集内蕴的拓扑结构突变影响, DMFSGD 算法的步长搜索次数也随之从 4 次左右跳变至 6~8 次; 而 ADCM 的表现则较为鲁棒, 步长搜索次数同样维持在 1.95~2.05 次之间, 与采用  $l_1$  损失函数时几乎相同, 比 DMFSGD 平均下降 50%~80%。如图 2(b)所示, 无论 DMFSGD 还是 ADCM 在历轮迭代最终选取的步长极为相似, 在两个不同的拓扑结构下, 分别为  $1 \times 10^{-4}$  和  $4 \times 10^{-5}$ , 说明 ADCM 在  $l_2$  损失函数被引入时对迭代步长的搜索能力也不弱于 DMFSGD。虽然 ADCM 的计算代价明显低于 DMFSGD, 重建精度和速度的表现却不因此而降低: 图 2(c)的 NPRE 演进曲线显示, ADCM 可以取得和 DMFSGD 几乎相同的收敛速度。可以看出, 由于此时不存在对误差的截断, 尽管搜索上界减小了一个数量级, 但相对于  $l_1$  损失函数,  $l_2$  损失函数的引入依然显著提升了 DMFSGD 和 ADCM 的收敛速度。值得注意的是, 伴随着收敛速度的提升, 矩阵重建的精度却不能尽如人意。如图 2(d)中的相对误差累积分布曲线所示,  $l_2$  损失函数作为评价准则时, 矩阵重建的误差显著高于  $l_1$  损失函数评价准则。基于前期工作的研究, 可认为是由于  $l_2$  损失函数对于时延序列中存在的抖动现象更加敏感, 难以处理随机延迟污染所致。

### 3.3 Huber 损失函数性能分析

Huber 损失函数是  $l_1$  损失函数和  $l_2$  损失函数的混合。实验中取  $\varepsilon = 15$  ms。实验证明, Huber 损失函数的引入能够兼顾收敛速度和重建精度。如图 3(a)所示, 基于 Huber 损失函数作为评价准则,

ADMC 的计算代价同样远远低于 DMFSGD。从算法运行的初始阶段开始直到稳定阶段, ADCM 的迭代搜索次数除最初的数轮迭代中上升到 2.1 次左右, 一直稳定在 1.95~2.05 次之间, 与其它损失函数被引入时几乎没有差异。相比之下, DMFSGD 的平均步长搜索次数从起初的 1 次攀升至稳定状态时的 6.5~6.8 次, 计算量远远高于 ADCM。图 3(b)中可以看出, 当系统处于稳定状态时, 虽然 ADCM 仅进行少数几次迭代, 但搜索得到的迭代步长与 DMFSGD 仍然十分接近: 虽然在起始阶段 ADCM 搜索得到的迭代步长约为 DMFSGD 的一倍, 但随着算法的收敛最终 ADCM 和 DMFSGD 都将梯度下降算法的迭代步长稳定在  $(2 \sim 2.5) \times 10^{-4}$  区间内, 这表明 ADCM 提出的步长搜索策略具备较强的泛化能力, 从侧面验证了策略的有效性。图 3(c)体现了算法的收敛速度。可以看出, 虽然 Huber 损失函数同样进行了误差截断, 但收敛速度却超过了采用  $l_1$  损失函数时的收敛速度。由于算法迭代初值位于原点附近, 因此绝大部分计算误差都落入  $l_1$  损失函数区间, 于是 Huber 损失函数的  $\varepsilon$  参数在此起到了步长放大的作用, 提升了收敛速度; 而当算法趋于稳定时, 相当部分的计算误差则落入  $l_2$  损失函数区间, 此时 Huber 损失函数截断了显著异常的计算误差, 从而有效过滤了时延毛刺所造成的延迟污染, 达到较好的重建精度。而从图 3(d)中可以看出, 此时矩阵重建的误差与  $l_1$  损失函数不相上下, 显著优于  $l_2$  损失函数。由以上实验可得, Huber 损失函数的引入能够兼顾收敛速度和异常数据处理能力, 具备优良的特性。

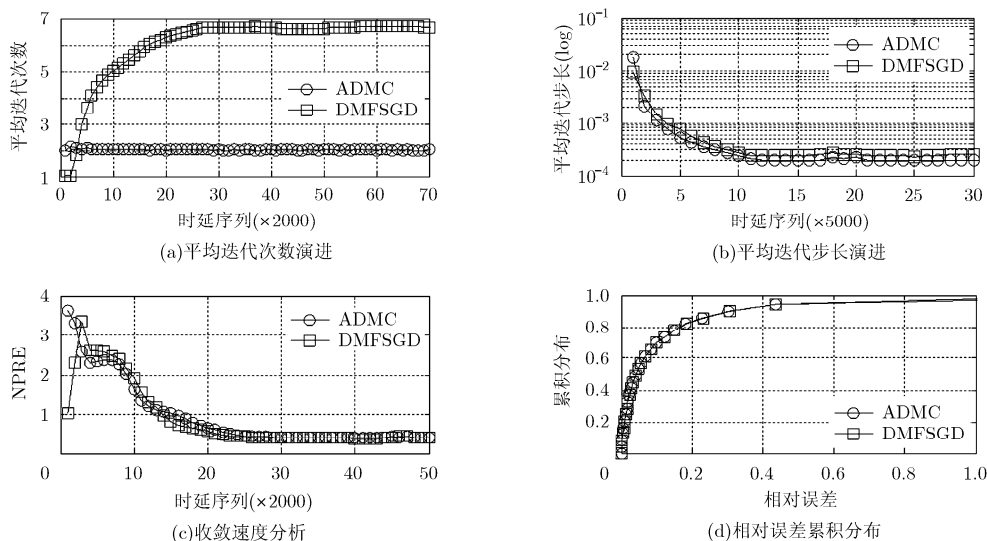


图3 Huber 损失函数算法性能

## 4 结束语

时延敏感型应用的优化强烈依赖于时延矩阵的感知与重建。不同于传统的分布式时延矩阵重建算法，本文在关注矩阵重建精度的同时，着重考虑了算法的计算代价和泛化能力，以确保实时性和特殊环境下的算法性能。由此，本文试图提出一种支持多种误差评价准则的时延矩阵重建算法：通过伸缩因子的引入来实现步长搜索上界的自适应升降，在满足不同评价准则和生命周期不同阶段下，梯度下降步长快速搜索的同时，以指数级的伸缩速率实现了对拓扑突变的敏捷适应。进一步地，本文引入了多种误差损失函数，在不损失计算精度的前提下，以较低的计算代价提升了模型的泛化能力。

## 参考文献

- [1] Szymaniak M, Presotto D, Pierre G, *et al.* Practical large-scale latency estimation[J]. *Computer Networks*, 2008, 52(7): 1343-1364.
- [2] Agarwal S and Lorch R. Matchmaking for online games and other latency-sensitive P2P systems[C]. Proceedings of the ACM SIGCOMM 2009, Barcelona, 2009: 315-326.
- [3] Armitage G and Heyde A. REED: optimizing first person shooter game server discovery using network coordinates[J]. *ACM Transactions on Multimedia Computing, Communications, and Applications*, 2012, DOI: 10.1145/21268996. 2169000.
- [4] Agarwal S, Dunagan J, Jain N, *et al.* Volley: automated data placement for geo-distributed cloud services[C]. Proceedings of the 7th USENIX Symposium on Networked Systems Design & Implementation (NSDI), San Jose, 2010: 1-16.
- [5] Ding C, Chen Y, Xu T, *et al.* CloudGPS: a scalable and ISP-friendly server selection scheme in cloud computing environments[C]. Proceedings on the 20th International Workshop on Quality of Service (IWQoS), Coimbra, 2012: 5.
- [6] 王占丰, 陈鸣, 邢长友, 等. 因特网延时空间建模的研究[J]. *通信学报*, 2012, 33(7): 164-176.
- [7] Wang Zhan-feng, Chen Ming, Xing Chang-you, *et al.* Research on the modeling of the Internet delay space[J]. *Journal on Communications*, 2012, 33(7): 164-176.
- [7] Lee S, Zhang Z L, Sahu S, *et al.* On suitability of Euclidean embedding for host-based network coordinate systems[J]. *IEEE/ACM Transactions on Networking*, 2010, 18(1): 27-40.
- [8] Candès E J and Plan Y. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements[J]. *IEEE Transactions on Information Theory*, 2011, 57(4): 2342-2359.
- [9] Candès E J and Recht B. Exact matrix completion via convex optimization[J]. *Foundations of Computational Mathematics*, 2009, 9(6): 717-772.
- [10] Mao Y, Saul L K, and Smith J M. IDES: an internet distance estimation service for large networks[J]. *IEEE Journal on Selected Areas in Communications*, 2006, 24(12): 2273-2284.
- [11] Wang Z F, Chen M, Xing C, *et al.* Multi-manifold model of the Internet delay space[J]. *Journal of Network and Computer Applications*, 2013, 31(1): 211-218.
- [12] Zhang R, Tang C, Hu Y C, *et al.* Impact of the inaccuracy of distance prediction algorithms on Internet applications-an analytical and comparative study[C]. Proceedings on 25th IEEE INFOCOM, Barcelona, 2006: 1-12.
- [13] Liao Y, Geurts P, and Leduc G. Network distance prediction based on decentralized matrix factorization[C]. Proceedings on the 9th International IFIP TC6 Networking Conference, Chennai, 2010: 15-26.

- [14] Chen Y, Wang X, Shi C, *et al.*. Phoenix: a weight-based network coordinate system using matrix factorization[J]. *IEEE Transactions on Network and Service Management*, 2011, 8(4): 334-347.
- [15] Tang L, Shen Z, Lin Q, *et al.*. Towards a robust framework of network coordinate systems[C]. Proceedings on the 9th International IFIP TC6 Networking Conference, Prague, 2012: 331-343.
- [16] Liao Y, Du W, Geurts P, *et al.*. DMFSGD: a decentralized matrix factorization algorithm for network distance prediction[J]. *IEEE/ACM Transactions on Networking*, 2013, 21(5): 1511-1524.
- [17] 史加荣, 焦李成, 尚凡华. 不完全非负矩阵分解的加速算法[J]. 电子学报, 2011, 39(2): 291-295.  
Shi Jia-rong, Jiao Li-cheng, and Shang Fan-hua. Accelerated algorithm to incomplete nonnegative matrix factorization[J]. *Acta Electronica Sinica*, 2011, 39(2): 291-295.
- [18] 张振跃, 赵科科. 数据缺损矩阵低秩分解的正则化方法[J]. 中国科学: 数学, 2013, 43(3): 249-271.
- Zhang Zhen-yue and Zhao Ke-ke. Regularization methods for low-rank factorization of matrices with missing data[J]. *SCIENTIA SINICA Mathematica*, 2013, 43(3): 249-271.
- [19] 胡毓达. 凸分析与非光滑分析[M]. 上海: 上海科学技术出版社, 2000: 52.
- [20] PlanetLabDataset[OL]. <http://www.eecs.harvard.edu/~syrah/nc/sim/pings.4hr.stamp.gz>. 2012.5.
- [21] Wu S, Chen Y, Fu X, *et al.*. NCSshield: securing decentralized, matrix factorization-based network coordinate systems[C]. Proceedings on the 20th International Workshop on Quality of Service (IWQoS), Coimbra, 2012: 1.
- 王 聪: 男, 1981 年生, 博士生, 研究方向为网络空间嵌入模型.  
张凤荔: 女, 1963 年生, 博士, 教授, 博士生导师, 研究方向为信息安全.  
王瑞锦: 男, 1980 年生, 博士生, 研究方向为无线网 3D 定位.